

EE 428: Computer Vision

Homework 1: RAW Processing

Instructor: Jonathan Ventura

In this homework you will implement a basic pipeline to convert the raw sensor output of a camera into a processed JPEG image.

The raw sensor output comes in the form of an array of 16-bit integer values, stored in a DNG file. To read the DNG file, we will use the `rawpy` library, which you can install by running the following command:

```
pip install rawpy
```

The values in the RAW file represent the per-pixel brightness readouts from a sensor with a Bayer camera filter array. The Bayer pattern is the standard pattern:

```
RGRG
GBGB
RGRG
GBGB
```

The file `hw1.ipynb` provides code to read a DNG file, using the `rawpy` library to read the file into a 16-bit integer NumPy array.

You will implement a basic version of the image post-processing software stack: demosaic, white balance, gamma curve, clip and quantize, and compress to JPEG.

Code requirements

1. Convert the raw image data to 32-bit floating point in $[0 \ 1]$ range.

Demosaic

Refer to the diagrams in the "Lecture 2.1: The Camera Pipeline" slides to help you understand this part.

2. Create the filter kernels you will need for the Bayer demosaicing.
3. Apply the filter kernels to the RAW image (using `scipy.ndimage.correlate`). Use `mirror` padding mode.
4. Create a `red`, `green`, and `blue` array, each of the same shape as the RAW image.
5. Copy the values from either the RAW image or the filter outputs as appropriate to form the interpolated red, green, and blue channel images. Use NumPy slicing and indexing, not for loops, to do this.
6. Stack the red, green, and blue channels into a single array.

White balance

7. Apply white balancing by scaling each channel so that its mean is 0.25. (Scaling means multiplying by a coefficient.)

Gamma curve

8. Apply an inverse gamma curve $x' = x^{1/\gamma}$ where $1/\gamma = 0.55$.

Clip and quantize

9. Clip to [0 1] range, scale by 255, and convert to 8-bit unsigned integer.

Compress

10. Save the image as a JPEG (using `imageio.imwrite`). The output filename is provided as `path_out`.

Report

Provide a short explanation of your solution. Be sure to document any sources you used in preparing your code, including websites and AI tools.

Submission instructions

Submit your Python notebook (.ipynb file) and report (PDF or docx). Please do not put them in a zip file, just submit the files directly.